

Disciplina:	RPG0025 – Lidando com sensores em dispositivos móveis
Nome:	Iggor Motta de Oliveira
Turma:	2025.4
Campus:	EAD
Curso:	Desenvolvimento Full stack
Link:	https://github.com/Iggor-motta/Mundo4-RPG0025

RPG0017 – Lidando com sensores em dispositivos móveis

Objetivo da Prática:

Nesta atividade a seguir compreenderemos que os apps do Wear OS podem funcionar como um dos principais frameworks para o desenvolvimento de aplicações mobile. Um aplicativo Wearable pode ter várias especialidades, desde entretenimento e comunicação.

Procedimentos adotados:

1º - Implementação de Saídas de áudio:

```
package com.doma.domawearsos
```

```
import android.content.Context
import android.media.AudioDeviceInfo
import android.content.pm.PackageManager
import android.media.AudioManager
```

```
class AudioHelper(private val appContext: Context) {
    private val audioManager =
        appContext.getSystemService(Context.AUDIO_SERVICE) as AudioManager
```

```

fun isAudioOutputAvailable(deviceType: Int): Boolean {
    val packageManager = appContext.packageManager
    if
(!packageManager.hasSystemFeature(PackageManager.FEATURE_AUDIO_O
UTPUT)) {
        return false
    }

    val devices =
audioManager.getDevices(AudioManager.GET_DEVICES_OUTPUTS)
    for (device in devices) {
        if (device.type == deviceType) {
            return true
        }
    }
    return false
}
}

```

2º - Detecção Dinâmica de Dispositivos de Áudio:

package com.doma.domawearsos

```

import android.content.Context
import android.content.pm.PackageManager
import android.media.AudioDeviceInfo
import android.media.AudioManager
import android.os.Bundle
import android.widget.Button
import android.widget.Toast
import androidx.appcompat.app.AlertDialog
import androidx.appcompat.app.AppCompatActivity

```

```

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val listButton = findViewById<Button>(R.id.addTaskButton)
        listButton.setOnClickListener { showAlertDialog() }

        val checkAudioButton = findViewById<Button>(R.id.button_check_audio)
        checkAudioButton.setOnClickListener { checkAudioOutput() }
    }
}

```

```

}

private fun showAlertDialog() {
    val builder = AlertDialog.Builder(this@MainActivity)
    builder.setTitle("Ação do Botão")
    builder.setMessage("Botão acionado!")

    builder.setPositiveButton(
        "OK"
    ) { dialog, which ->

    }

    val dialog = builder.create()
    dialog.show()
}

```

3º - Facilitando a Conexão Bluetooth:

```

private fun checkAudioOutput() {
    val audioHelper = AudioHelper(this)

    val isSpeakerAvailable =

audioHelper.audioOutputAvailable(AudioDeviceInfo.TYPE_BUILTIN_SPEAKER
)
    val isBluetoothHeadsetConnected =

audioHelper.audioOutputAvailable(AudioDeviceInfo.TYPE_BLUETOOTH_A2D
P)

    Toast.makeText(this, "Alto-falante disponível: $isSpeakerAvailable",
Toast.LENGTH_SHORT)
        .show()
    Toast.makeText(
        this,
        "Fone de ouvido Bluetooth conectado: $isBluetoothHeadsetConnected",
Toast.LENGTH_SHORT
    ).show()
}

```

4º - Reprodução de Áudio e o Uso de Alto-falantes em Dispositivos Wear OS:

```
internal inner class AudioHelper(private val context: Context) {
    private val audioManager = context.getSystemService(AUDIO_SERVICE)
    as AudioManager

    fun audioOutputAvailable(type: Int): Boolean {
        if
        (!context.packageManager.hasSystemFeature(PackageManager.FEATURE_AUDIO_OUTPUT)) {
            return false
        }
        val devices =
        audioManager.getDevices(AudioManager.GET_DEVICES_OUTPUTS)
        for (device in devices) {
            if (device.type == type) {
                return true
            }
        }
        return false
    }
}
```

Códigos solicitados na missão encontra-se neste link
<https://github.com/Iggor-motta/Mundo4-RPG0025>